

Adabas

The *Com-plete Application Programmer's documentation* describes the call interface to Adabas. This call interface provided by Com-plete deals exclusively with Com-plete timing, rolling and statistics. The Adabas linkage is provided by separate ADALNK modules.

This chapter covers the following topics:

- General Usage
 - ADALNK Features
-

General Usage

All Adabas calls are issued in the key of the thread itself. As there is a requirement to sometimes roll out a user over an Adabas call, the Adabas interface requires that it can get its UB area and a certain amount of its workarea outside of the thread and in the same key as the thread. For this reason, an Adabas buffer pool is built with subpools of the same size but in different storage protect keys.

The size of the Adabas workarea is dependent on the various lengths set in the ADALCO module when it is assembled. It is possible to have a user length and a Review length included in the calculation for the Adabas workarea in this way. The total amount of the Adabas workarea should be rounded up to a good binary figure such as 512 or 1024. This is because the buffer subpools must be page aligned and without a value such as one of these, storage will be wasted.

If no Adabas buffer pool is specified by the user, Com-plete allocates a buffer pool with a subpool in each user storage protect key possible with 8k available in each. The number of Adabas work areas that can be allocated from each will be dependent on it's total length. It is not possible for Com-plete to make a more intelligent guess than this as their usage totally depends on the number of Adabas calls which will be issued concurrently in a given storage protect key.

It is recommended that users calculate their Adabas usage and build their Adabas buffer pool accordingly. You must build a buffer pool with a subpool available in every key in which an Adabas call will be issued. If an Adabas call is issued in a key for which no subpool exists, the subpool will be created but this in itself is an additional overhead. Software AG recommend that a subpool be created for each key in which an Adabas call can be issued in the Com-plete region.

You must now determine how many Adabas work areas will be required at the same time in each subpool. Software AG recommend that the resultant figure be increased by 20% to allow for peaks and future use. If the subpool fills, it will expand if the virtual storage is available but again, this is an overhead. If the subpools are well allocated and their usage monitored closely, it should be possible to increase or decrease the specification as required.

The Adabas work areas will be allocated above the line when the following conditions are satisfied. If one of the following is not satisfied, the Adabas buffer subpools will be allocated below the line.

ADALCO must be linked with AMODE=31

If ULOPADAB exists it must be linked with AMODE=31

There must be no Adabas version 4 usage in the system

ADALNK Features

The V5 ADALNK features of user exits and data passing are supported by the Com-plete interface:

- **User exits**

If required, you can link the UEXITB and UEXITA routines with the supplied ADALNK module. The ADALNK routine is called in the AMODE in which the MCALL Adabas was issued, this also applies to UEXITB and UEXITA. You must ensure that the return is made in the correct mode, that is, in the mode in which he was entered. Please note that parameters passed from Natural can reside above the 16 MB line.

- **Data passing**

The ADALNK module is also supplied in source so that you can re-assemble the module providing a value for LUINFO, the UB extension length. This length is added to the basic length of the workarea (approx. 512 bytes) and the total is the size of the area obtained by the Com-plete nucleus from the buffer pool. This means that careful adjustment of the BUFFERPOOL sysparm may be necessary to prevent inefficient use of storage.

A full description of these features is contained in the Adabas documentation.